# **Experiment 4:**

# **Combinational Logic Design with MSI**

# **Magnitude Comparator, Decoders and Multiplexers**

**Objective**

* To practice more with combining small decoders to construct a bigger one.
* To practice the easiness in designing and implementing different output functions with MSI components like Decoders & multiplexers.
* Compare different ways of design using MSI & SSI components.

**Basic Concepts**

Numerous MSI devices are available commercially that perform specific digital functions commonly employed in the design of digital systems. The selection of MSI components rather than SSI gates is extremely important since, in most cases, it will result in a considerable reduction of IC packages and interconnecting wires. In this experiment, we will use the magnitude comparator, decoder and the multiplexer in implementing some designs.

**I - Magnitude Comparator**

The comparison of two numbers is an operation that determines whether one number is greater than, equal to, or less than the other number. Two numbers A and B can be compared by subtracting B from A. and then:

* If the result is equal to zero, then A is equal to B.
* If the result is positive, then A is greater than B.
* If the result is negative, then A is less than B.

A 4-bit magnitude comparator is available as a standard MSI chip (74LS85). This component takes two 4-bit number A=(A3A2A1A0) and B=(B3B2B1B0) as inputs and produces three outputs each corresponding to one of the following conditions: (OA<B), (OA>B) and (OA=B). There are also three extra inputs (IA<B), (IA>B), and (IA=B) used to enable the designer to cascade more than one 74LS85 in order to build comparators that compare numbers that are more than 4-bits long. The pin diagram for a 74LS85 and the corresponding truth table are shown in figure 1 next page.



**74LS85**



Figure 1: Pin diagram and truth table of a 74LS85

**II - Decoders**

A decoder is any circuit that detects the presence of unique input states or condition and yields only one active output for that condition. The decoder converts the binary information present on "n" input lines to a maximum of 2n unique output lines. For each possible input condition, one and only one output signal will be active. A n-to-2n decoder can also be considered as a "minterm generator". In general, an n-to-m line decoder has "n" inputs and "m" outputs. n and m should satisfy m ≤ 2n. The block diagram for a 3-to-8 line decoder and the corresponding truth table are shown in Figure 2.

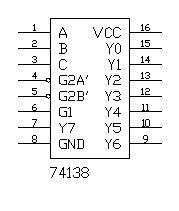
**3X8**

# DECODER

Truth table of a 3-to-8 line decoder

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INPUTS | | | OUTPUTS | | | | | | | |
| x | y | z | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

The 3-to-8 decoder is available as a standard TTL chip (74138). The pin diagram and the truth table for 74138 chips are shown in Figure 2. Note that the outputs are active low for the 74138. G1, G2A', and G2B' are the enable inputs, for proper operation G1 = 1, and G2A' = G2B' = 0.



FUNCTION TABLE

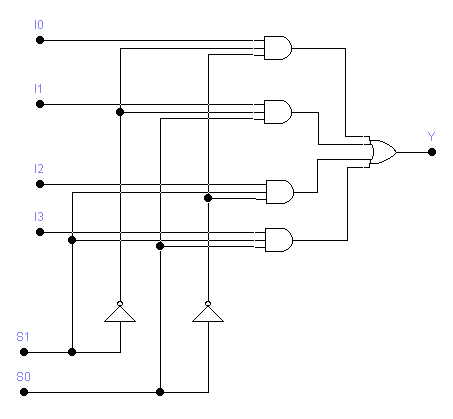
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ENABLE  OUTPUTS | | SELECT  INPUTS | | | OUTPUTS | | | | | | | |
| X | H | X | X | X | H | H | H | H | H | H | H | H |
| L | X | X | X | X | H | H | H | H | H | H | H | H |
| H | L | L | L | L | L | H | H | H | H | H | H | H |
| H | L | L | L | H | H | L | H | H | H | H | H | H |
| H | L | L | H | L | H | H | L | H | H | H | H | H |
| H | L | L | H | H | H | H | H | L | H | H | H | H |
| H | L | H | L | L | H | H | H | H | L | H | H | H |
| H | L | H | L | H | H | H | H | H | H | L | H | H |
| H | L | H | H | L | H | H | H | H | H | H | L | H |
| H | L | H | H | H | H | H | H | H | H | H | H | L |

Any Boolean function can be expressed in sum of min-terms form. Therefore, a decoder can be used to generate the min-terms. Then, an external OR gate can form the sum of the desired min-terms and implement the desired function. Since the outputs of the 74138 are active low, it is preferable to replace the OR gate with a NAND gate.

**III - Multiplexers**

A digital multiplexer (MUX) is a combinational circuit that selects binary information from one of its many input lines and directs it to its single output line. This selection is controlled by a set of selection lines. Normally, there are up to 2n input lines when the multiplexer has n selection lines.

For example, a 4-to-1 line MUX is shown in Figure 2 next page. Each of the four input lines I0 to I3 is applied to one input of an AND gate. Selection lines S1 and S0 are decoded to select a particular AND gate (e.g. if the S1 S0 = 10 the I2 input would be the output of the multiplexer). A multiplexer is also called a *data selector*, since it selects one of many inputs and steers the binary information to the output line.



|  |  |  |
| --- | --- | --- |
| *S1* | *S0* | *Y* |
| 0 | 0 | *I0* |
| 0 | 1 | *I1* |
| 1 | 0 | *I2* |
| 1 | 1 | *I3* |

Function Table

Block Diagram

### 4x1 MUX

### Inputs

### Select

I0

I1

I2

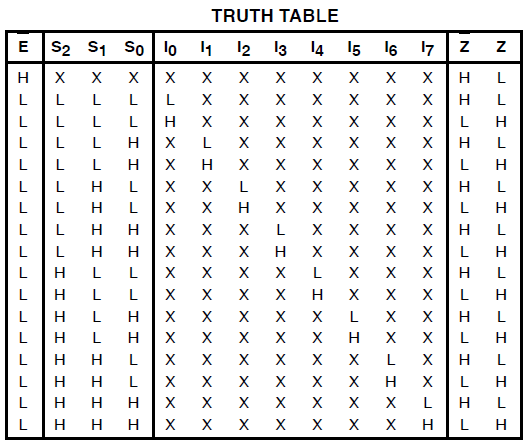
I3

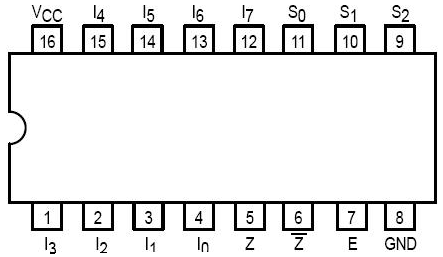
### Output

### S1 S0

Y

Figure 2: Block diagram and Function table of a 4×1 MUX

An 8-to-1 line MUX is available as a standard MSI chip (74LS151). The chip comes in a 16-pin package. The select inputs are named S2, S1, and S0. Enable inputs are active low. Output is provided in both active-high and active-low versions. The 74LS151 is shown in figure 3.



**74LS151**

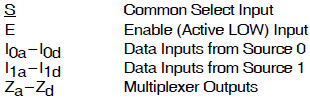
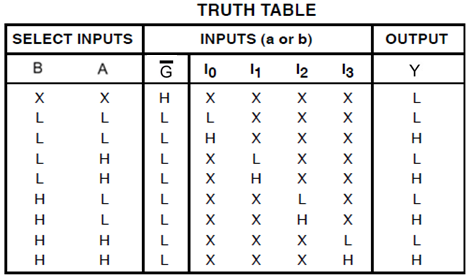


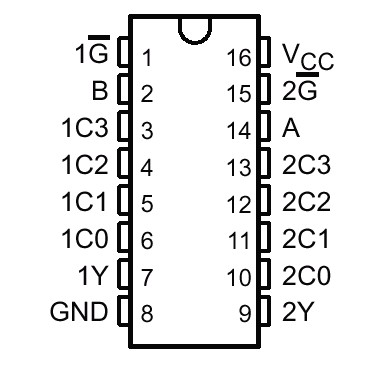
Figure 3: Block diagram and Function table of a 4×1 MUX

The 74LS153 is a Dual 4-Input Multiplexer with common select inputs and individual enable inputs for each section. It can select two bits of data from four sources. The two outputs present data in the true (non-inverted) form. The 74LS153 is shown in figure 4.



**74LS153**

**Logic Symbol**



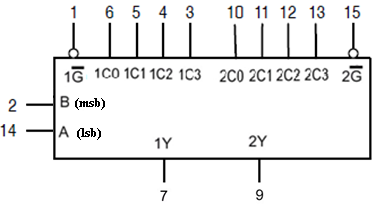
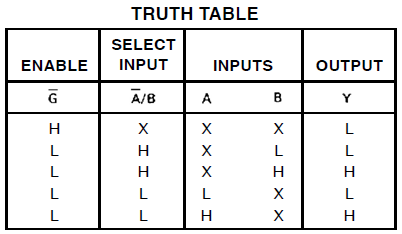
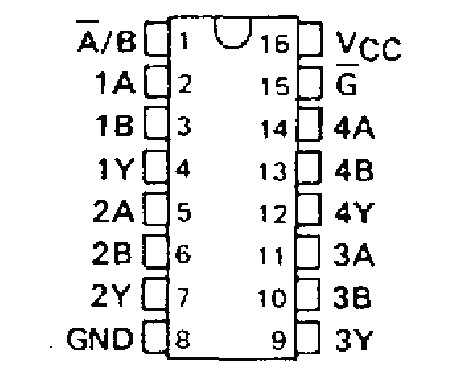


Figure 4: Dual 4-to-1 MUX

The 74LS157 is a Quad 2-Input Multiplexer. Four bits of data from two sources can be selected using the common Select and Enable inputs. The four outputs present the selected data in the true (non-inverted) form. The 74LS157 is shown in figure 5.



**Logic Symbol**

**74LS157**

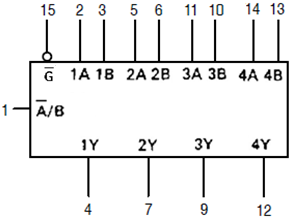


Figure 5: Quad 2-to-1 MUX

A designer can implement any Boolean function of *n* variables using a 2*n-1* input multiplexer. A 4-variable function can be implemented using a 8-to-1 MUX by using three of the variables as select inputs and feeding an appropriate function of the fourth variable (or 0, or 1) to the data inputs of the MUX.

**Procedure**

**Part 1:**

# Simulate using EWB then implement on lab board the following Boolean function using 74LS151 and a single inverter with variable z as its input:

***F(w, x, y, z) = Σ m (0, 2 , 8, 10)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***w*** | ***x*** | ***y*** | ***z*** | ***F*** | ***MUX Input*** |
| 0 | 0 | 0 | 0 | 1 | Z’ |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | Z’ |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | Z’ |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | Z’ |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

# Simulate using EWB then implement on lab board the same Function above using I C 74LS138 and 2 input NAND gates in I C 7400.

* Simulate using EWB then implement on lab board the same function above using any of the 4x16 decoder you designed in experiment #3 and any other gates you used before at the lab.
* Simulate using EWB then implement on lab board the same function above using any other gates you used before at the lab.
* Compare the above four ways in designing the same function in terms of ease of design and number of I.C.’s used in board implementation.

**Part 3:**

You have two numbers, namely A and B, each of 4 bits.

**Complete on paper only** the circuit shown in figure 6 below to display the larger or equal number using a 7-segment display as below or as what you have at EWB.

13

12

11

10

9

15

14

2

6

1

7

16

8

12

CC

Vcc

MAN74A

GND

A

B

C

D

a

b

c

d

e

f

g

**7447**

47 Ω

330 Ω

330 Ω

330 Ω

330 Ω

330 Ω

330 Ω

330 Ω

*a*

*b*

*c*

*d*

*e*

*f*

*g*

14

13

8

7

6

1

2

a

b

c

d

e

f

g

**Magnitude**

**Comparator**

**A**

**B**

OA>B

OA<B

OA=B

IA>B

IA<B

IA=B

?

?

?

**Quad**

**2-to-1 MUX**

**A**

**B**

**Y**

G’

A / B

?

?

?

Figure 6: Block diagram of part 3 circuit